

# «Подсистема энтропийного кодирования при сжатии информации»

# Схема работы GDCT кодека



# Сжатие информации

```
graph TD; A[Сжатие информации] --- B[Методы Сжатия информации]; B --- C[С потерями]; B --- D[Без потерь];
```

Методы  
Сжатия  
информации

С потерями

Без потерь

# Стратегии сжатия

- *Статистическая стратегия* сжатия предполагает определение вероятностей элементов.
- В *блочных* методах статистика элементов отдельно кодируется и добавляется к сжатому блоку.
- В *поточных* (адаптивных) методах вычисление вероятностей для элементов поступающих данных производится на основе априорных вероятностей из предыдущих данных.
- *Преобразующая стратегия* не предполагает вычисления вероятностей. Результат преобразования имеет лучшую структуру данных и может быть сжат простым и быстрым методом

# Классификация методов сжатия

|  | Статистические        |                                | Преобразующие                   |           |
|--|-----------------------|--------------------------------|---------------------------------|-----------|
|  | Поточные              | Блочные                        | Поточные                        | блочные   |
| Модель<br><i>«источник<br/>без<br/>памяти»</i><br>(поток<br>элементов) | Адаптивный<br>HUFFMAN | Статистический<br>HUFFMAN      | SEM, VQ,<br>MTF, DC,<br>SC, DWT | DFT, DCT  |
| Модель<br><i>«источник с<br/>памятью»</i><br>(поток слов)              | CM, DMC,<br>PPM       | CMBZ, precon-<br>ditioned PPMZ | LZ*                             | BWT, ST   |
| Модель<br>элементов<br>или битов                                       | Адаптивный<br>ARIC    | Статистический<br>ARIC         | RLE, LPC                        | PBS, ENUC |

# Расшифровка названий методов

- CM (Context modeling) – контекстное моделирование.
- DMC (Dynamic Markov compression) – динамическое марковское сжатие
- (частный случай CM)
- PPM (Predictio by partial match) – предсказание по частичному совпадению (частный случай CM).
- LZ\* (LZ77, LZ78, LZH, LZW) – методы Зива – Лемпеля.
- HUFFMAN ( Huffman coding) – кодирование Хаффмана.
- RLE (Run length encoding) – кодирование длин повторов.
- SEM (Separate exponents and mantissas) – разделение экспонент и мантисс
- (представление целых чисел).
- UNIC (Universal coding) – универсальное кодирование
- (частный случай SEM).
- ARIC (Arithmetic coding) – арифметическое кодирование.
- RC (Range coding) – интервальное кодирование
- (вариант арифметического кодирования).

# Расшифровка названий методов

- DC (Distance coding) – кодирование расстояний.
- IF (Inversion frequencies) – «обратные частоты» (вариант DC).
- MTF (Move to front) – «сдвиг к вершине», «перемещение стопки книг».
- ENUC (Enumerate coding) – нумерующее кодирование.
- DFT (Discrete Fourier transform) – ДПФ- дискретное преобразование Фурье
- DCT (Discrete cosine transform) – ДКП – дискретное косинусное преобразование
- DWT (Discrete wavelet transform) – дискретное вейвлет – преобразование.
- LPC ( Linear prediction coding) – кодер линейного предсказания.
- PBS (Parallel blocks sorting) – сортировка параллельных блоков.
- ST (Sort transformation) – частичное сортирующее преобразование
- (частный случай PBS)
- BWT (Burrows – Wheeler transform) – преобразование Барроуза – Уиллера (частный случай ST)

# Характеристики сжатия

- а) *фактор сжатия*  $r = FS/F0$ ,
- б) *коэффициент сжатия*  $k = F0/FS = 1/r$ ,
- в) *качество сжатия*  
 $\eta = 100(1-r) = 100(F0-FS)/FS$ .
- Здесь  $F0$  и  $FS$  – размеры исходного и выходного (сжатого) файлов).
- Очевидно, что при  $r < 1$ ,  $k > 1$  происходит сжатие выходного файла. Параметр  $\eta < 100$  показывает относительное уменьшение в процентах сжатого файла по сравнению с исходным файлом.



Энтропия сообщения по К.Шеннону – bps (bit per symbol)

$$H = - \sum_{i=1}^N P_i \cdot \log_2(P_i)$$

## Теоретический предел длины сжатого сообщения

$$L' = n \cdot H$$

$$L' = -n \sum_{i=1}^N P_i \cdot \log_2(P_i)$$

$$L = n \cdot l,$$

*Длина символа в битах*

*число символов*

# Пример расчета энтропии сообщения длины сжатого сообщения и коэффициента сжатия

- Сообщение :
- *Длинношеее животное*
- Частоты символов

|        |   |   |   |   |   |   |   |   |   |   |   |
|--------|---|---|---|---|---|---|---|---|---|---|---|
| symbol | д | л | и | н | о | ш | е | ж | в | т |   |
| $k_i$  | 1 | 1 | 2 | 3 | 3 | 1 | 4 | 1 | 1 | 1 | 1 |

- Число символов  $n=19$
- Энтропия  $H=3.221$  bps
- Длина сжатого сообщения  $L'=n \cdot H=61.201$  bit
- Длина исходного сообщения  $L=n \cdot 8=152$  bit
- Коэффициент сжатия  $k=2.484$

# Пример расчета энтропии сообщения длины сжатого сообщения и коэффициента сжатия

- Сообщение :
- *2718281828*
- Частоты СИМВОЛОВ

|        |   |   |   |   |
|--------|---|---|---|---|
| symbol | 2 | 7 | 1 | 8 |
| $k_i$  | 3 | 1 | 2 | 4 |

- Число СИМВОЛОВ  $n=10$
- Энтропия  $H=1.846$  bps
- Длина сжатого сообщения  $L'=n \cdot H=18.46$  bit
- Длина исходного сообщения  $L=n \cdot 8=80$  bit
- Коэффициент сжатия  $k=4.33$

# Некоторые методы сжатия без потерь

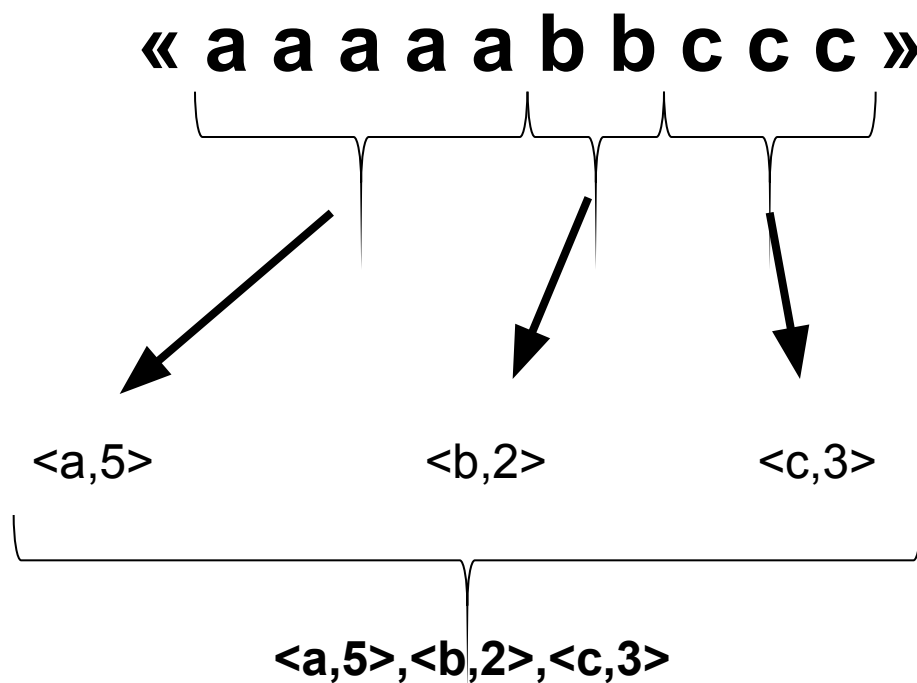
Энтропийное кодирование

**Кодирование Хаффмана**

**Арифметическое кодирование**

**Кодирование длин  
непрерывных  
Последовательностей  
(RLE)**

# Кодирование длин непрерывных последовательностей (RLE)



# Алгоритм кодирования Хаффмана

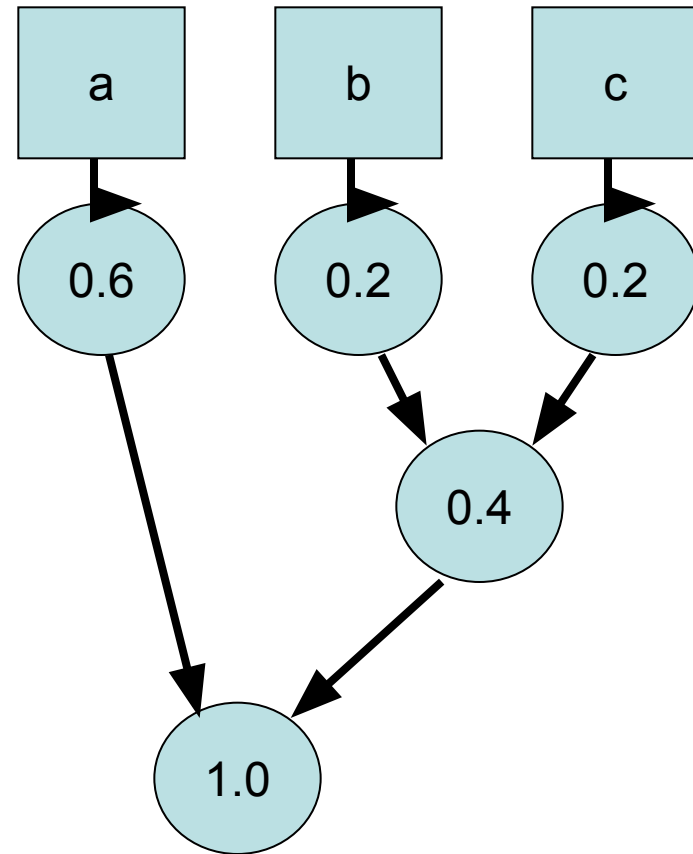
«aabc» = 00 00 01 00 10  
10 бит

Таблица вероятностей:

| a   | b   | c   |
|-----|-----|-----|
| 00  | 01  | 10  |
| 0.6 | 0.2 | 0.2 |

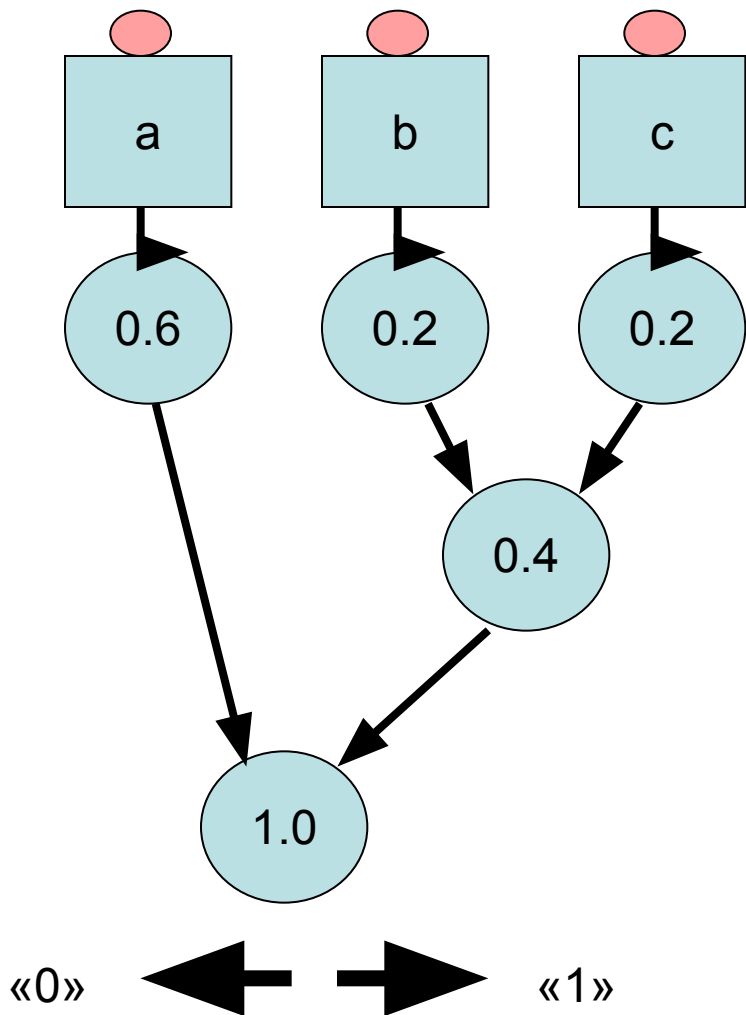
$L' = 6.855$  бит

## Построение дерева Хаффмана



# Кодирование Хаффмана

## Дерево Хаффмана

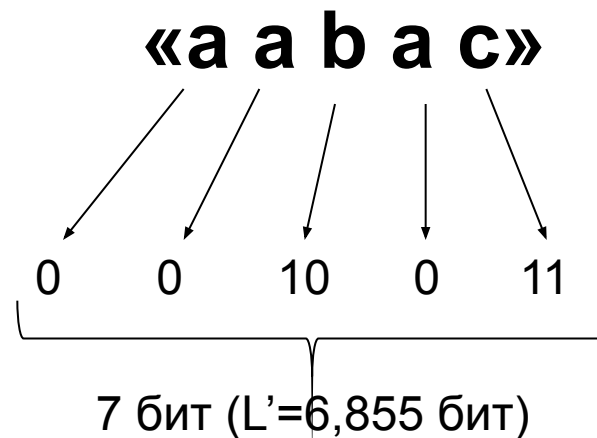


## Коды Хаффмана

'a' - «0»

'b' - «10»

'c' - «11»



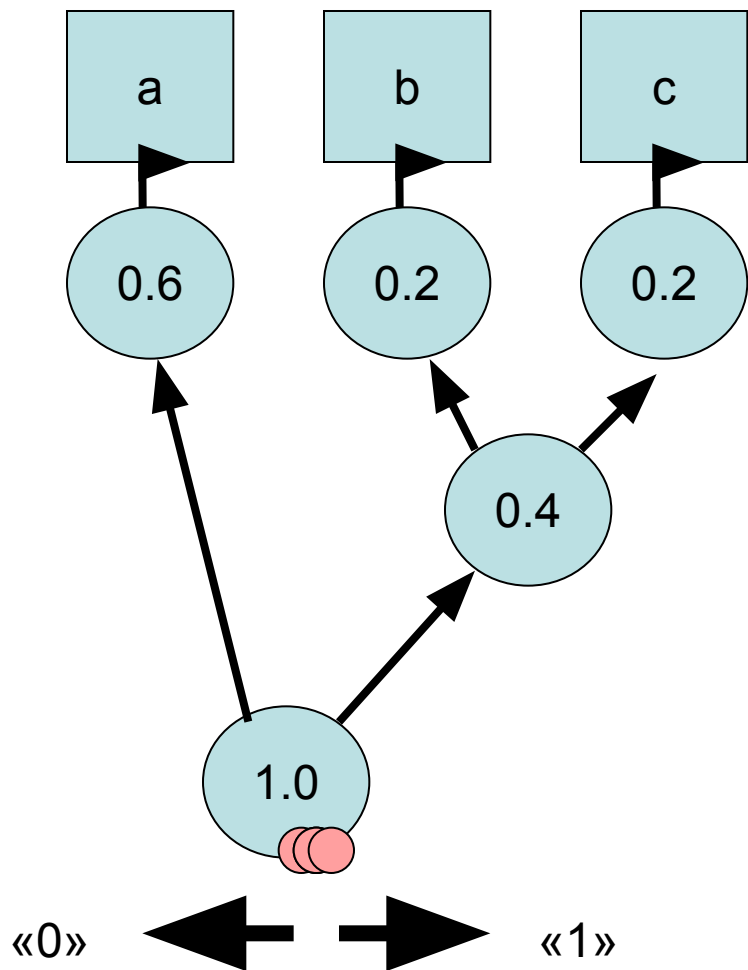
0 - движение по левой ветви

1 - движение по правой ветви



# Декодирование Хаффмана

## Дерево Хаффмана



Последовательность  
кодов Хаффмана:

**0 0 1 0 0 1 1**

0 - a

0 - a

1 0 - b

0 - a

1 1 - c

Сообщение восстановлено

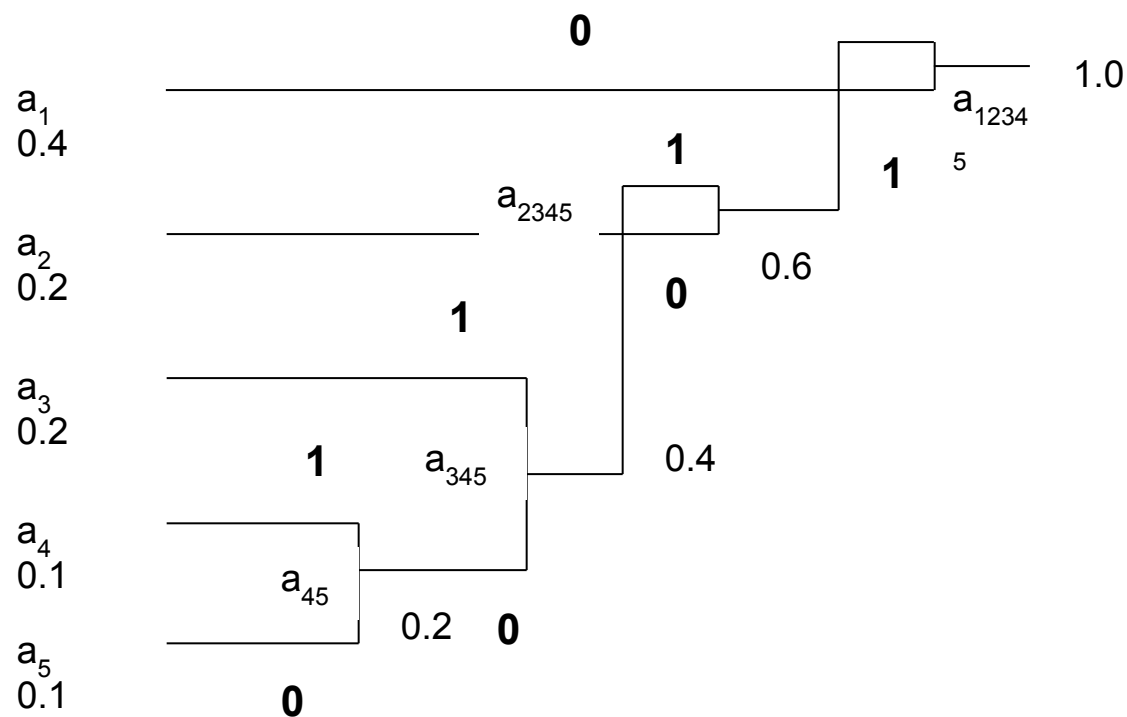
**«a a b a c»**

0 - движение по левой ветви

1 - движение по правой ветви

# Дерево Хаффмана – 1 вариант

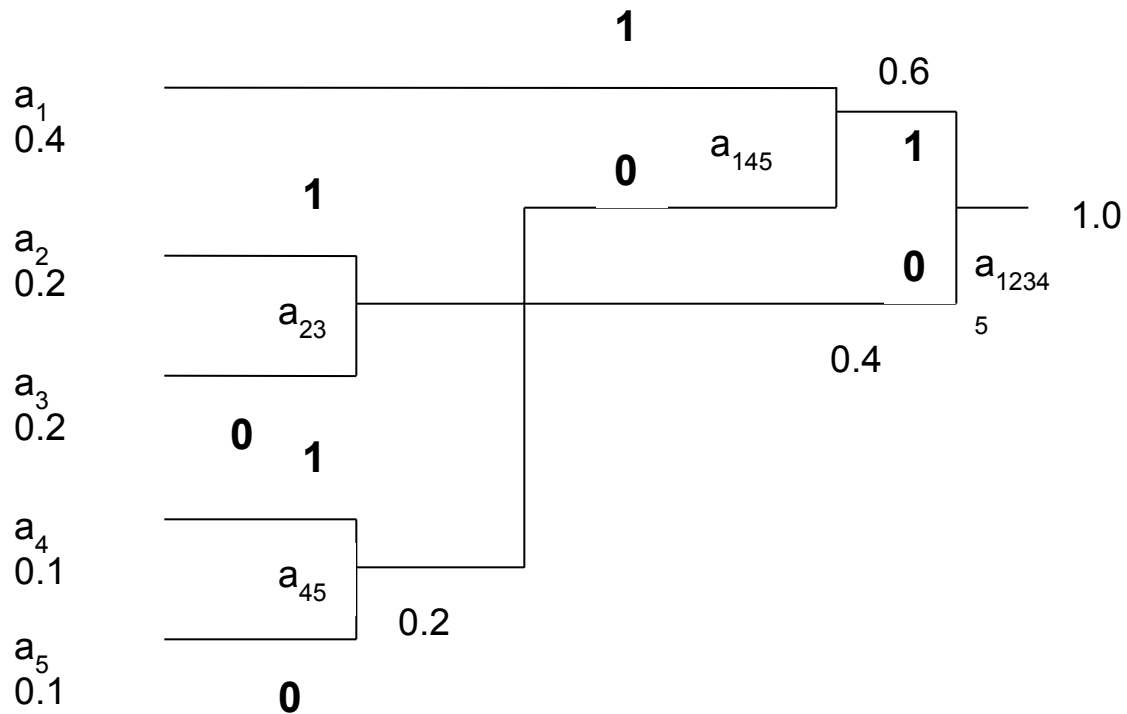
- Код
- 0
- 10
- 111
- 1101
- 1100



- Энтропия  $H=2.2$  bps дисперсия длин кодов 1.36

# Дерево Хаффмана – 2 вариант

- Код
- 11
- 01
- 00
- 101
- 100



- Энтропия  $H=2.2$  bps, Дисперсия длин кодов 0.16

# Дисперсия длин кодов

- Средняя длина кода
- $l_i$  - длина  $i$ -го кода в битах
- Дисперсия кода
- В стандартах используют готовые коды VLC (коды переменной длины)

$$H = -\sum_{i=1}^N P_i \cdot \log_2(P_i)$$

$$D = \sum_{i=1}^n p_i (l_i - H)^2$$

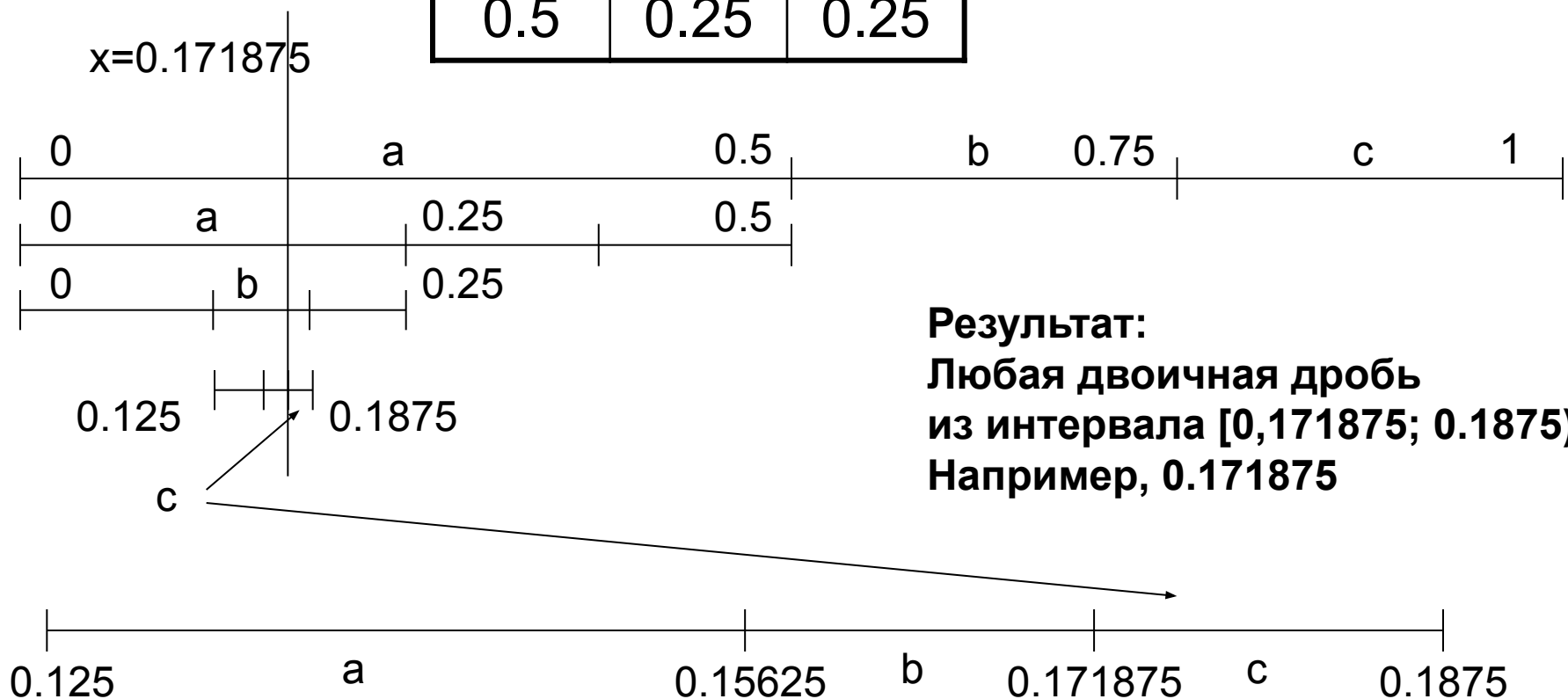
# Арифметическое кодирование

«aabc»

Таблица вероятностей:

| a   | b    | c    |
|-----|------|------|
| 0.5 | 0.25 | 0.25 |

$L'=6$



## Арифметическое кодирование

**«aabc»**

Таблица вероятностей:

| a   | b    | c    |
|-----|------|------|
| 0.5 | 0.25 | 0.25 |

**Результат:**

**Любая двоичная дробь**

**из интервала [0,171875; 0.1875)**

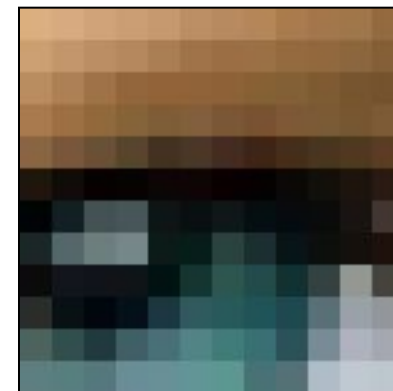
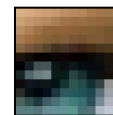
**Например, 0.171875**

Выходной поток:  $0.171875d=0.001011b=$ «001011»

# Сравнение кодов

- VLC – удобны для реализации, не универсальны, средняя длина отлична от энтропии
- Двухпроходные коды Хаффмана – средняя длина кода близка к энтропии, информация о дереве должна присоединяться к сжатой информации.
- Просты в программировании
- Адаптивные коды Хаффмана не требуют априорных сведений о вероятностях символов, компрессор и декомпрессор должны быть идентичными
- Арифметический кодер – средняя длина кода практически равна энтропии. Достаточно сложен в программировании. Требования априорных сведений, как у кода Хаффмана. Дерево кодирования-декодирования однозначно

## Исходное изображение «Masha»





## Результат восстановления



RLE: 14953 байт (сжатие: 25,29)

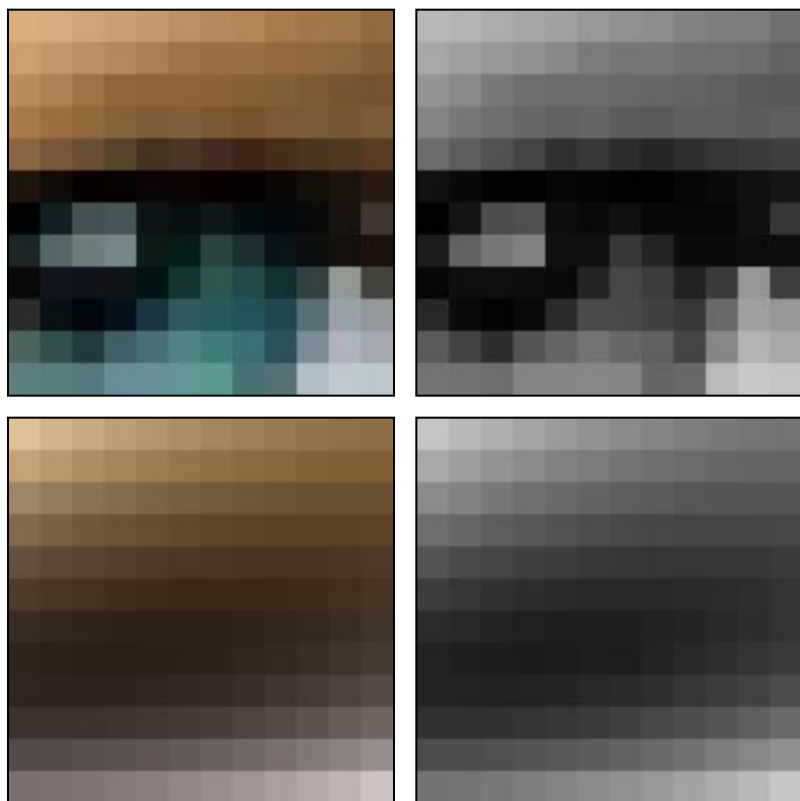
RLE+Huffman: 11047 байт (сжатие: 34,24)

RLE+Arithm: 11022 байт (сжатие: 34,32)

**PSNR(Y)=20,578 дБ, сжатие: 34,32**



## Результат восстановления



RLE: 5929 байт (сжатие: 63,80)

RLE+Huffman: 4229 байт (сжатие: 89,44)

RLE+Arithm: 4209 байт (сжатие: 89,87)

**PSNR(Y)=19,752 дБ, сжатие: 89,87**

